



OPEN

Cellular automaton decoders for topological quantum codes with noisy measurements and beyond

Michael Vasmer^{1,2,3✉}, Dan E. Browne¹ & Aleksander Kubica^{2,3}

We propose an error correction procedure based on a cellular automaton, the sweep rule, which is applicable to a broad range of codes beyond topological quantum codes. For simplicity, however, we focus on the three-dimensional toric code on the rhombic dodecahedral lattice with boundaries and prove that the resulting local decoder has a non-zero error threshold. We also numerically benchmark the performance of the decoder in the setting with measurement errors using various noise models. We find that this error correction procedure is remarkably robust against measurement errors and is also essentially insensitive to the details of the lattice and noise model. Our work constitutes a step towards finding simple and high-performance decoding strategies for a wide range of quantum low-density parity-check codes.

Developing and optimizing decoders for quantum error-correcting codes is an essential task on the road towards building a fault-tolerant quantum computer^{1–3}. A decoder is a classical algorithm that outputs a correction operator, given an error syndrome, i.e. a list of measurement outcomes of parity-check operators. The error threshold of a decoder tells us the maximum error rate that the code (and hence an architecture based on the code) can tolerate. Moreover, decoder performance has a direct impact on the resource requirements of fault-tolerant quantum computation⁴. In addition, the runtime of a decoder has a large bearing on the clock speed of a quantum computer and may be the most significant bottleneck in some architectures^{5,6}.

Here, we focus on decoders for CSS stabilizer codes^{7,8}, in particular topological quantum codes^{9–12}, which have desirable properties such as high error-correction thresholds and low-weight stabilizer generators. Recently, there has been renewed interest in d -dimensional topological codes, where $d \geq 3$, because they have more powerful logical gates than their two-dimensional counterparts^{13–21} and are naturally suited to networked^{22–25} and ballistic linear optical architectures^{26–30}. In addition, it has recently been proposed that one could utilize the power of three-dimensional (3D) topological codes using a two-dimensional layer of active qubits^{31,32}.

Cellular-automaton (CA) decoders for topological codes^{33–39} are particularly attractive because they are local: at each vertex of the lattice we compute a correction using a simple rule that processes syndrome information in the neighbourhood of the vertex. This is in contrast to more complicated decoders such as the minimum-weight perfect matching algorithm¹⁰, which requires global processing of the entire syndrome to compute a correction. Moreover, CA decoders have another advantage: they exhibit single-shot error correction^{40,41}, i.e. it is not necessary to repeat stabilizer measurements to compensate for the effect of measurement errors.

In our work we study the recently proposed sweep decoder³⁹, a cellular-automaton decoder based on the sweep rule. First, we adapt the sweep rule to an abstract setting of codes without geometric locality, which opens up the possibility of using the sweep decoder for certain low-density parity-check (LDPC) codes beyond topological codes. Second, we show how the sweep decoder can be used to decode phase-flip errors in the 3D toric code on the rhombic dodecahedral lattice with boundaries, and we prove that it has a non-zero error threshold in this case. We remark that the original sweep decoder only works for the toric code defined on lattices without boundaries. Third, we numerically simulate the performance of the decoder in the setting with measurement errors and further optimize its performance. We use an independent and identically distributed (iid) error model with phase-flip probability p and measurement error probability q . We observe an error threshold of $\sim 2.1\%$ when $q = p$, an error threshold of $\sim 2.9\%$ when $q = 0$ and an error threshold of $\sim 8\%$ when $p \rightarrow 0^+$; see

¹Department of Physics and Astronomy, University College London, Gower Street, London WC1E 6BT, UK. ²Perimeter Institute for Theoretical Physics, Waterloo, ON N2L 2Y5, Canada. ³Institute for Quantum Computing, University of Waterloo, Waterloo, ON N2L 3G1, Canada. ✉email: mvasmer@perimeterinstitute.ca

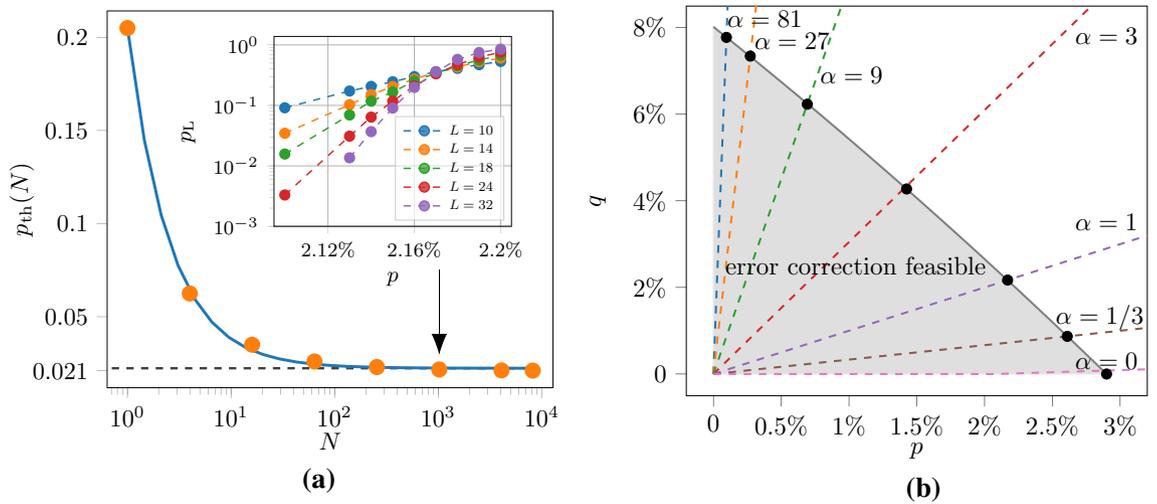


Figure 1. Numerical error threshold estimates for the sweep decoder applied to the toric code on the rhombic dodecahedral lattice. In (a), we plot the error threshold $p_{th}(N)$ as function of the number of error-correction cycles N , for an error model with equal phase-flip (p) and measurement error (q) probabilities ($\alpha = q/p = 1$). The inset shows the data for $N = 2^{10}$, where we use 10^4 Monte Carlo samples for each point. Using the ansatz in Eq. (10), we estimate the sustainable threshold to be $p_{sus} \approx 2.1\%$. In (b), we plot p_{sus} for error models with different values of α , where we approximate $p_{sus} \approx p_{th}(2^{10})$.

Decoder	$q = 0$	$q = p$
Neural network ⁴⁹	17.5%	N/A
Renormalization group (RG) ⁵⁰	17.2%	7.3%
Toom’s rule ⁴⁸	14.5%	N/A
Erasure mapping (EM) ⁵¹	12.2%	N/A
Sweep (with direction change)	15.5%	1.7%

Table 1. Comparison of the error thresholds of (cubic) toric code decoders against phase-flip noise with ($q = p$) and without ($q = 0$) measurement errors. Our results are shown in bold.

Fig. 1. We note that the sweep rule decoder cannot be used to decode bit-flip errors in the 3D toric code, as their (point-like) syndromes do not have the necessary structure. However, 2D toric code decoders can be used for this purpose, e.g.^{10,42–47}.

We report that the sweep decoder has an impressive robustness against measurement errors and in general performs well in terms of an error-correction threshold. To compare the sweep decoder with previous work^{48–51}, we look at the 3D toric code on the cubic lattice, as decoding the 3D toric code on the rhombic dodecahedral lattice has not been studied before; see Table 1.

The remainder of this article is structured as follows. We start by presenting how the sweep rule can be used in an abstract setting of codes without geometric locality. Then, we outline a proof of the non-zero error threshold of the sweep decoder for the 3D toric code on the rhombic dodecahedral lattice with boundaries. We also present numerical simulations of the performance of the decoder in the setting with measurement errors for lattices with and without boundaries. We discuss the applicability of the sweep decoder and suggest directions for further research. Finally, we prove the properties of the sweep rule in the abstract setting, and analyze the case of lattices with boundaries.

Results

We start this section by adapting the sweep rule³⁹ to the setting of causal codes, which go beyond topological quantum codes. Then, we focus on the 3D toric code on the rhombic dodecahedral lattice. We first analyze the case of the infinite lattice, followed by the case of lattices with boundaries. We finish by presenting numerical simulations of the performance of various optimized versions of the sweep decoder.

Sweep rule for causal codes. Recall that a stabilizer code is CSS iff its stabilizer group can be generated by operators that consist exclusively of Pauli X or Pauli Z operators. Let \mathcal{Q} denote the set of physical qubits of the code and \mathcal{S} be the set of all X stabilizer generators, which are measured. We refer to the stabilizers returning -1 outcome as the X -type syndrome. The X -type syndrome constitutes the classical data needed to correct Pauli Z errors. In what follows, we focus on correcting Z errors as X errors are handled analogously.

We start by introducing a partially ordered set V with a binary relation \preceq over its elements. We refer to the elements of V as locations. Given a subset of locations $U \subseteq V$, we say that a location $w \in V$ is an upper bound of U and write $U \preceq w$ iff $u \preceq w$ for all $u \in U$; a lower bound of U is defined similarly. The supremum of U , denoted by $\sup U$, is the least upper bound of U , i.e., $\sup U \preceq w$ for each upper bound w of U . Similarly, the infimum $\inf U$ is the greatest lower bound of U . We also define the future and past of $w \in V$ to be, respectively

$$\uparrow(w) = \{W \subseteq V : w \preceq W\}, \tag{1}$$

$$\downarrow(w) = \{W \subseteq V : W \preceq w\}. \tag{2}$$

We define the causal diamond

$$\diamond(U) = \uparrow(\inf U) \cap \downarrow(\sup U) \tag{3}$$

of U as the intersection of the future of $\inf U$ and the past of $\sup U$. Lastly, for any $\mathcal{A} \subseteq 2^V$, where 2^V is the power set of V , we define the restriction of \mathcal{A} to the location $v \in V$ as follows

$$\mathcal{A}|_v = \{A \in \mathcal{A} : A \ni v\}. \tag{4}$$

For notational convenience, we use the shorthands $\bigcup \mathcal{A} = \bigcup_{A \in \mathcal{A}} A$ and $\sup \mathcal{A} = \sup \bigcup \mathcal{A}$.

Let $C_{\mathcal{Q}}$ and $C_{\mathcal{S}}$ be \mathbb{F}_2 -linear vector spaces with the sets of qubits \mathcal{Q} and X stabilizer generators \mathcal{S} as bases, respectively. Note that there is a one-to-one correspondence between vectors in $C_{\mathcal{Q}}$ and subsets of \mathcal{Q} , thus we treat them interchangeably; similarly for vectors in $C_{\mathcal{S}}$ and subsets of \mathcal{S} . Let $\partial : C_{\mathcal{Q}} \rightarrow C_{\mathcal{S}}$ be a linear map, called the boundary map, which for any Pauli Z error with support $\epsilon \subseteq \mathcal{Q}$ returns its X -type syndrome $\sigma \subseteq \mathcal{S}$, i.e., $\sigma = \partial\epsilon$. We say that a location $v \in V$ is trailing for $\sigma \in \text{im } \partial$ iff $\sigma|_v$ is nonempty and belongs to the future of v , i.e., $\sigma|_v \subset \uparrow(v)$.

Now, we proceed with defining a causal code. We say that a quadruple $((V, \preceq), \mathcal{Q}, \mathcal{S}, \partial)$ describes a causal code iff the following conditions are satisfied.

1. (causal diamonds) For any finite subset of locations $U \subseteq V$ there exists the causal diamond $\diamond(U)$.
2. (locations) Every qubit $Q \in \mathcal{Q}$ and every stabilizer generator $S \in \mathcal{S}$ correspond to finite subsets of locations, i.e., $Q, S \subseteq V$.
3. (qubit infimum) For every qubit $Q \in \mathcal{Q}$ its infimum satisfies $\inf Q \in Q$.
4. (syndrome evaluation) The syndrome $\partial\epsilon$ of any error $\epsilon \subseteq \mathcal{Q}$ can be evaluated locally, i.e.,

$$\forall v \in V : (\partial\epsilon)|_v = [\partial(\epsilon|_v)]|_v. \tag{5}$$

5. (trailing location) For any location $v \in V$ and the syndrome $\sigma \in \text{im } \partial$, if $\sigma|_v$ is nonempty and $\sigma|_v \subset \uparrow(v)$, then there exists a subset of qubits $\varphi(v) \subseteq \mathcal{Q}|_v \cap \uparrow(v)$ satisfying $[\partial\varphi(v)]|_v = \sigma|_v$ and $\diamond(\varphi(v)) = \diamond(\sigma|_v)$.

We can adapt the sweep rule to any causal code $((V, \preceq), \mathcal{Q}, \mathcal{S}, \partial)$. We define the sweep rule for every location $v \in V$ in the same way as in Ref.³⁹.

Definition 1 (sweep rule) If v is trailing, then find a subset of qubits $\varphi(v) \subseteq \mathcal{Q}|_v \cap \uparrow(v)$ with a boundary that locally matches σ , i.e. $[\partial\varphi(v)]|_v = \sigma|_v$. Return $\varphi(v)$.

Our first result is a lemma concerning the properties of the sweep rule. But, before we state the lemma, we must make some additional definitions. Let $u, v \in V$ be two locations satisfying $u \prec v$. We say that a sequence of locations $u \prec w_1 \prec \dots \prec w_n \prec v$, where $n = 0, 1, \dots$, forms a chain between u and v of length $n + 1$. We define $\mathcal{N}(u, v)$ to be the collection of all the chains between u and v . We write $\ell(N)$ to denote the length of the chain $N \in \mathcal{N}(u, v)$.

Given a causal code $((V, \preceq), \mathcal{Q}, \mathcal{S}, \partial)$, we define its corresponding syndrome graph G as follows. For each stabilizer $S \in \mathcal{S}$, there is a node in G and we add an edge between any two nodes iff their corresponding stabilizers both have a non-zero intersection with the same qubit in \mathcal{Q} . We define the syndrome distance $d_G(S, T)$ between any two stabilizer generators $S, T \in \mathcal{S}$ to be the graph distance in G , i.e., the length of the shortest path in G between the nodes corresponding to S and T . This can be extended to the syndromes $\sigma, \tau \subseteq \mathcal{S}$ in the obvious way: $d_G(\sigma, \tau) = \min_{S \in \sigma, T \in \tau} d_G(S, T)$.

Lemma 1 (sweep rule properties) Let $\sigma \in \text{im } \partial$ be a syndrome of the causal code $((V, \preceq), \mathcal{Q}, \mathcal{S}, \partial)$. Suppose that the sweep rule is applied simultaneously at every location in V at time steps $T = 1, 2, \dots$, and the syndrome is updated at each time step as follows: $\sigma^{(T+1)} = \sigma^{(T)} + \partial\varphi^{(T)}$, where $\varphi^{(T)}$ is the set of qubits returned by the rule. Then,

1. (support) the syndrome at time T , $\sigma^{(T)}$ stays within the causal diamond of the original syndrome, i.e.

$$\sigma^{(T)} \subseteq \diamond(\sigma), \tag{6}$$

2. (propagation) the syndrome distance between σ and any $S \in \sigma^{(T)}$ is at most T , i.e.

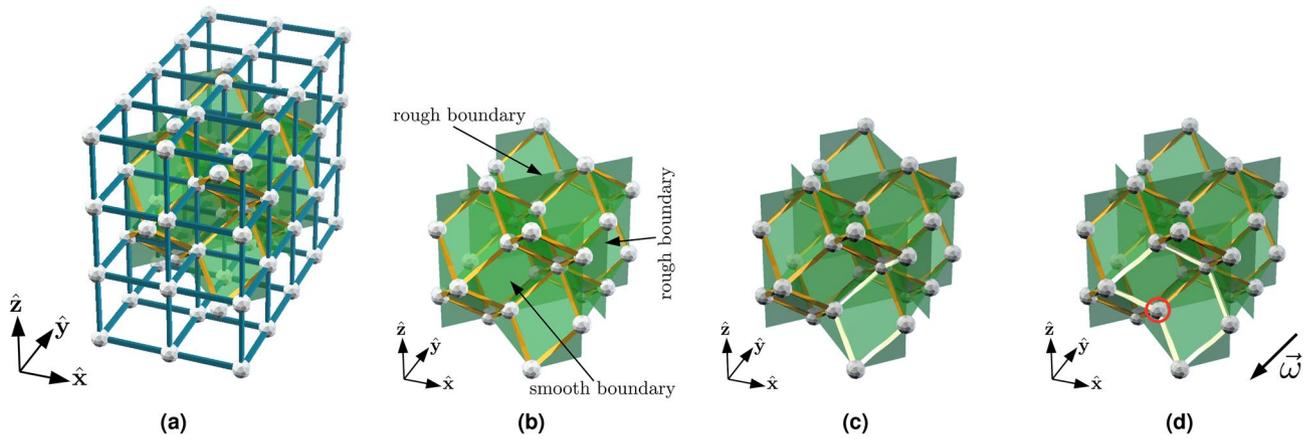


Figure 2. A family of rhombic dodecahedral lattices with boundaries. **(a)** Construction of the $L = 3$ lattice from the cubic lattice. We show the initial cubic lattice as well as the final rhombic dodecahedral lattice. **(b)** The $L = 3$ lattice. The front and back boundaries are smooth (syndromes must form closed loops) and the other boundaries are rough (open loops of syndrome can terminate). **(c)** An example of an open loop of syndrome terminating on one of the rough boundaries (light yellow edges). **(d)** A syndrome on the smooth boundary (light yellow edges). With the sweep direction $\vec{\omega} = -(1, 1, 1)$, the ringed vertex does not satisfy the trailing location condition.

$$d_G(S, \sigma) \leq T, \tag{7}$$

3. (removal) the syndrome is trivial, i.e. $\sigma^{(T)} = 0$, for

$$T > \max_{v \in \bigcup \sigma} \max_{N \in \mathcal{N}(v, \sup \sigma)} \ell(N). \tag{8}$$

That is, the syndrome is trivial for times T greater than the maximal chain length between a location v and the supremum of the syndrome $\sup \sigma$, maximized over all locations v contained in the syndrome (viewed as a subset of locations).

The above lemma is analogous to Lemma 2 from Ref.³⁹, but is not limited to codes defined on geometric lattices; rather, it now applies to more general causal codes. We defer the proof until the Methods.

Rhombic dodecahedral toric codes. In this section, we examine an example of a causal code: the 3D toric code defined on the infinite rhombic dodecahedral lattice. Toric codes defined on this lattice are of interest because they arise in the transversal implementation of CCZ in 3D toric codes^{16,18,32}. Consider the tessellation of \mathbb{R}^3 by rhombic dodecahedra, where a rhombic dodecahedron is a face-transitive polyhedron with twelve rhombic faces. One can construct this lattice from the cubic lattice, as follows. Begin with a cubic lattice, where the vertices of the lattice are the elements of \mathbb{Z}^3 . Create new vertices at all half-integer coordinates $(x/2, y/2, z/2)$, satisfying $x + y + z \equiv 1 \pmod{4}$ and $xyz \equiv 1 \pmod{2}$. These new vertices sit at the centres of half of all cubes in the cubic lattice. For each such cube, add edges from the new vertex at its centre to the vertices of the cube. Finally, delete the edges of the cubic lattice. The remaining lattice is a rhombic dodecahedral lattice. Figure 2 gives an example of this procedure.

We denote the infinite rhombic dodecahedral lattice by \mathcal{L}^∞ , and we denote its vertices, edges, faces, and cells by $\mathcal{L}_0^\infty, \mathcal{L}_1^\infty, \mathcal{L}_2^\infty$, and \mathcal{L}_3^∞ respectively. We place qubits on faces, and we associate X and Z stabilizer generators with edges and cells, respectively. That is, for each edge $e \in \mathcal{L}_1^\infty$, we have a stabilizer generator $\prod_{f:e \in f} X_f$, and for each cell $c \in \mathcal{L}_3^\infty$, we have a stabilizer generator $\prod_{f \in c} Z_f$, where X_f (Z_f) denotes a Pauli X (Z) operator acting on the qubit on face f . In the notation of the previous section, $V = \mathcal{L}_0^\infty, \mathcal{S} = \mathcal{L}_1^\infty$, and $\mathcal{Q} = \mathcal{L}_2^\infty$. Let $\vec{\omega} \in \mathbb{R}^3$ be a vector (a sweep direction) that is not perpendicular to any of the edges of \mathcal{L}^∞ . Such a sweep direction induces a partial order over \mathcal{L}_0^∞ , as we now explain. Let $(u : v)$ denote a path from one vertex $v \in \mathcal{L}_0^\infty$ to another vertex $u \in \mathcal{L}_0^\infty$, where $(u : v) = \{(u, w_1), \dots, (w_n, v)\}$ is a set of edges. We call a path from u to v causal (denoted by $(u \downarrow v)$) if the inner product $\vec{\omega} \cdot (w_i, w_{i+1})$ has the same sign for all edges $(w_i, w_{i+1}) \in (u \downarrow v)$. We write $u \leq v$, if $u = v$ or there exists a causal path, $(u \downarrow v)$, and $\vec{\omega} \cdot (w_i, w_{i+1}) > 0$ for all edges in the path.

We now verify that the rhombic dodecahedral toric code equipped with a sweep direction is a causal code. We note that this is only with respect to phase-flip errors, as for bit-flip errors the trailing location condition is not satisfied. Higher-dimensional toric codes can satisfy the causal code conditions for both bit-flip and phase-flip errors, e.g. the 4D toric code with qubits on faces^{10,52}. We choose a sweep direction that is parallel to one of the edge directions of the lattice, $\vec{\omega} = (1, 1, 1)$. First, consider the causal diamond condition. One can prove by induction that any finite subset of vertices of the lattice has an infimum and supremum, and therefore has a unique causal diamond. Figure 3 shows an example of the future of a vertex and the causal diamond of a subset of vertices. By definition, the qubits and stabilizer generators are associated with faces and edges, which are finite subsets of locations (vertices), so the locations condition is satisfied. Next, consider the qubit infimum condition. The faces of the lattice are rhombi, and as $\vec{\omega}$ is not perpendicular to any of the edges of the lattice, each face contains its infimum.

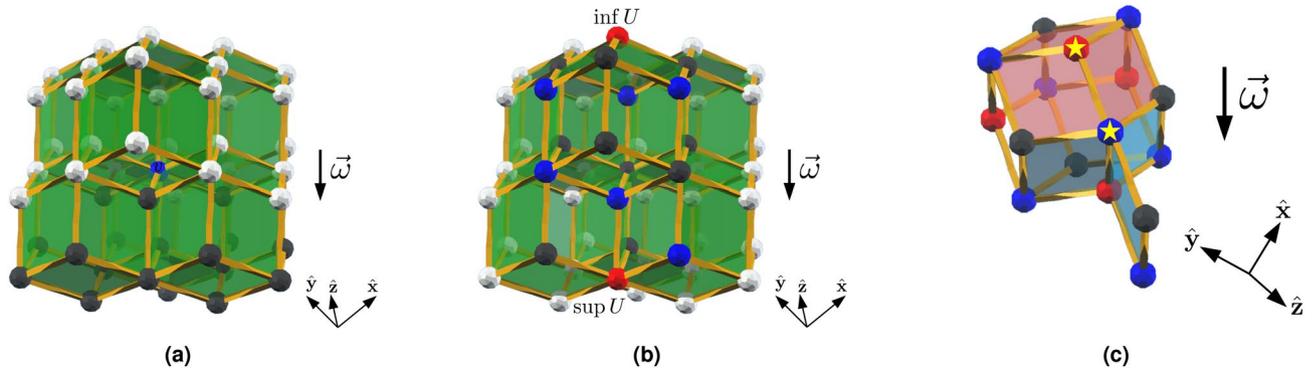


Figure 3. The sweep rule in the rhombic dodecahedral lattice. The sweep direction $\vec{\omega} = -(1, 1, 1)$ is indicated by the arrow. (a) The future $\uparrow(v)$ of v (the blue vertex), is the power set of the black vertices and the blue vertex. (b) The causal diamond of U , where U is the set of blue vertices. The red vertices are $\text{sup } U$ and $\text{inf } U$, and $\diamond(U)$ is the power set of the red, black and blue vertices. (c) In the rhombic dodecahedral lattice, there are two types of vertices: one type is degree four (red and black) and the other is degree eight (blue). The red (blue) shaded faces are the qubits that the rule may return, depending on the syndrome at the highlighted red (blue) vertex. The rule returns nothing at the black vertices because there are no syndromes whose restriction to a black vertex is in the future of that vertex.

Next, consider the syndrome evaluation condition. We consider vector spaces C_1 and C_2 with bases given by $e \in \mathcal{L}_1^\infty$ and $f \in \mathcal{L}_2^\infty$, respectively. This allows us to define the linear boundary operator $\partial : C_2 \rightarrow C_1$, which is specified for all basis elements $f \in C_2$ as follows: $\partial f = \sum_{e \in f} e$. In words, for each face, ∂ returns the sum of the edges in the face. The syndrome of a (phase-flip) error $\epsilon \subseteq \mathcal{L}_2^\infty$ is then $\partial \epsilon$. This syndrome evaluation procedure is local.

Now, we consider the final causal code condition: the trailing location condition. The rhombic dodecahedral lattice has two types of vertex: one type is degree four and the other is degree eight. Given our choice of sweep direction, one can easily verify that the trailing location condition is satisfied at all the vertices of \mathcal{L}^∞ , as illustrated in Fig. 3c.

Sweep decoder for toric codes with boundaries. In this section, we consider the extension of the sweep rule to 3D toric codes defined on a family of rhombic dodecahedral lattices \mathcal{L} with boundaries, with growing linear size L . We discuss the problems that occur when we try to apply the standard sweep rule to this lattice. Then, we present a solution to these problems in the form of a modified sweep decoder.

We construct a family of rhombic dodecahedral lattices with boundaries from the infinite rhombic dodecahedral lattice, by considering finite regions of it. We start with a cubic lattice of size $(L - 1) \times (L + 1) \times L$, with vertices at integer coordinates $(x, y, z) \in [0, L - 1] \times [0, L + 1] \times [0, L]$. To construct a rhombic dodecahedral lattice, we create vertices at the centre of cubes with coordinates $(x/2, y/2, z/2)$ satisfying $x + y + z = 1 \pmod 4$ and $xyz = 1 \pmod 2$. Then for each vertex at the centre of a cube, we create edges from this vertex to all the vertices (x, y, z) of its cube satisfying $0 < y < L + 1$ and $0 < z < L$. Finally, we delete the edges of the cubic lattice and all the vertices (x, y, z) of the cubic lattice with $y = 0, L + 1$ or $z = 0, L$. Figure 2 illustrates the construction of the rhombic dodecahedral lattice \mathcal{L} for $L = 3$.

We denote the vertices, edges, and faces of the lattice \mathcal{L} by $\mathcal{L}_0, \mathcal{L}_1$, and \mathcal{L}_2 , respectively. As in the previous section, we associate qubits with the faces of the lattice and we associate X stabilizer generators with the edges. We define a local region of \mathcal{L} to be a region of \mathcal{L} with diameter smaller than $L/2$. In particular, there are no non-trivial logical Z operators supported within a local region. The boundary map ∂ is defined in the same way as the infinite case, except for an important caveat. Some faces in \mathcal{L} only have one or two edges (see Fig. 2), so ∂ can return the sum of fewer than four edges. In the bulk of the lattice, syndromes must form closed loops. As illustrated in Fig. 2b, the lattices in our family have two types of boundaries: rough and smooth. Open loops of syndrome may only terminate on the rough boundaries.

For a chosen sweep direction, some vertices on the boundaries of \mathcal{L} will not satisfy the trailing location condition (see Fig. 2d for an example). This means that some syndromes near the boundary are immobile under the action of the Sweep rule. Clearly, this poses a significant problem for a decoder based on the Sweep rule, as there are some constant weight errors whose syndromes are immobile. Fortunately, there is a simple solution to this problem: periodically varying the sweep direction. We pick a set of eight sweep directions

$$\Omega = \{(\pm 1, \pm 1, \pm 1)\}, \tag{9}$$

where each $\vec{\omega} \in \Omega$ is parallel to one of the eight edge directions of \mathcal{L} . The rule for each direction is analogous to that shown in Fig. 3c.

For our set of sweep directions, there are no errors within a local region whose syndromes are immobile for every sweep direction. To make this statement precise, we need to adapt the definition of a causal diamond to lattices with boundaries. As \mathcal{L} is a subset of the infinite rhombic dodecahedral lattice \mathcal{L}^∞ , the causal diamond of $U \subseteq \mathcal{L}_0$ is well defined in \mathcal{L}^∞ for any $\vec{\omega} \in \Omega$, but it may contain subsets of vertices that are not in \mathcal{L} . Roughly

speaking, we define the causal region of U with respect to $\vec{\omega}$, $\mathcal{R}_{\vec{\omega}}(U)$, to be the restriction of the causal diamond in \mathcal{L}^∞ (with respect to $\vec{\omega}$) to the finite lattice \mathcal{L} . More precisely, it is a subset of the elements of the causal diamond, whose vertices belong to \mathcal{L} , i.e. $\mathcal{R}_{\vec{\omega}}(U) = 2^{\diamond_{\vec{\omega}}(U) \cap \mathcal{L}_0}$, where $\diamond_{\vec{\omega}}(U)$ is the causal diamond with respect to $\vec{\omega}$. We now state a lemma that is sufficient to prove that a decoder based on the Sweep rule has a non-zero error threshold for rhombic dodecahedral lattices with boundaries.

Lemma 2 *Let Ω be a set of sweep directions and $\sigma \in \text{im } \partial$ be a syndrome, such that, for all sweep directions $\vec{\omega} \in \Omega$, the causal region of σ with respect to $\vec{\omega}$, $\mathcal{R}_{\vec{\omega}}(\sigma)$, is contained in a local region of \mathcal{L} . Then there exists a sweep direction $\vec{\omega}^* \in \Omega$ such that the trailing location condition is satisfied at every vertex in $\mathcal{R}_{\vec{\omega}^*}(\sigma)$.*

We defer the proof of Lemma 2 until the Methods. We now give a pseudocode description of a modified sweep decoder that works for 3D toric codes defined on lattices with boundaries. We note that all addition below is carried out modulo 2, as we view errors and syndromes as \mathbb{F}_2 vectors.

Algorithm 1 SWEEP DECODER

Input: Error syndrome $\sigma \in \text{im } \partial$, set of sweep directions Ω , runtime per direction T_{max}

Output: Correction $\varphi \subseteq \mathcal{L}_2$

$\varphi \leftarrow 0$

for $j \leftarrow 1$ **to** $|\Omega|$ **do**

$T \leftarrow 1$

$\vec{\omega} \leftarrow \Omega[j]$

▷ Change sweep direction.

while $T \leq T_{max}$ **do**

Apply the sweep rule with sweep direction $\vec{\omega}$ simultaneously to every vertex of \mathcal{L} to get $\varphi^{(T)} \subseteq \mathcal{L}_2$, the set of qubits returned by the rule.

$\sigma \leftarrow \sigma + \partial \varphi^{(T)}$

▷ Update syndrome.

$\varphi \leftarrow \varphi + \varphi^{(T)}$

▷ Update correction.

$T \leftarrow T + 1$

if $\sigma \neq 0$ **then**

return FAIL

else

return φ

The decoder can fail in two ways. Firstly, if the syndrome is still non-trivial after $|\Omega| \times T_{max}$ applications of the sweep rule, we consider the decoder to have failed. Secondly, the decoder can fail because the product of the correction and the original error implements a non-trivial logical operator. We can now state our main theorem.

Theorem 1 *Consider 3D toric codes defined on a family of rhombic dodecahedral lattices \mathcal{L} , with growing linear size L . There exists a constant $p_{th} > 0$ such that for any phase-flip error rate $p < p_{th}$, the probability that the sweep decoder fails is $O\left((p/p_{th})^{\beta_1 L^{\beta_2}}\right)$, for some constants $\beta_1, \beta_2 > 0$.*

We provide a sketch of the proof here, and postpone the details until Supplementary Note 1. Our proof builds on^{33,39,45,53} and relies on standard results from the literature^{45,54}.

Proof First, we consider a chunk decomposition of the error. This is a recursive definition, where the diameter of the chunk is exponential in the recursion level. Next, we use the properties of the sweep rule (Lemma 1) to show that the sweep decoder successfully corrects chunks up to some level $m^* = O(\log L)$. To accomplish this, we first show that the sweep decoder is guaranteed to correct errors whose diameter is smaller than L , in a number of time steps that scales linearly with the diameter. Secondly, we rely on a standard lemma that states that a connected component of a level- m chunk is well separated from level- n chunks with $n \geq m$, which means that the sweep decoder corrects connected components of the error independently. Finally, percolation theory tells us that the probability of an error containing a level- n chunk is $O\left((p/p_{th})^{2^n}\right)$, for some $p_{th} > 0$. As the decoder successfully corrects all level- n chunks for $n < m^* = O(\log L)$, the failure probability of the decoder is $O\left((p/p_{th})^{\beta_1 L^{\beta_2}}\right)$. □

Numerical implementation and optimization. We implemented the sweep decoder in C++ and simulated its performance for 3D toric codes defined on rhombic dodecahedral lattices, with and without boundaries. The code is available online⁵⁵. We study the performance of the sweep decoder for error models with phase-flip and measurement errors. Specifically, we simulate the following procedure.

Definition 2 (*Decoding with noisy measurements*) Consider the 3D toric code with qubits on faces and X stabilizers on edges. At each time step $T \in \{1, \dots, N\}$, the following events take place:

1. A Z error independently affects each qubit with probability p .
2. The X stabilizer generators are measured perfectly.

- Each syndrome bit is flipped independently with probability $q = \alpha p$, where α is a free parameter.
- The sweep rule is applied simultaneously to every vertex, and Z corrections are applied to the qubits returned by the rule.

After N time steps have elapsed, the X stabilizer generators are measured perfectly and we apply Algorithm 1. Decoding succeeds if, and only if, the product of the errors and corrections (including the correction returned by Algorithm 1) is a Z -type stabilizer.

We note that measuring the stabilizers perfectly after N time steps may seem unrealistic. However, this can model readout of a CSS code, because measurement errors during destructive single-qubit X measurements of the physical qubits have the same effect as phase-flip errors immediately prior to the measurements.

We first present our results for lattices with periodic boundary conditions i.e. each lattice is topologically a 3-torus. Although changing the sweep direction is not necessary for such lattices, we observe improved performance compared with keeping a constant sweep direction. We consider error models with phase-flip error probability p and measurement error probability $q = \alpha p$, for various values of α . For a given error model, we study the error threshold of the decoder as a function of N , the number of cycles, where a cycle is one round of the procedure described in Definition 2. For a range of values of N , we estimate the logical error rate, p_L , as a function of p for different values of L (the linear lattice size, or equivalently the code distance of the toric code). We estimate the error threshold as the value of p at which the curves for different L intersect. We find that the error threshold decays polynomially in N to a constant value, the sustainable threshold. The sustainable threshold is a measure of how much noise the decoder can tolerate over a number of cycles much greater than the code distance. We use the following numerical ansatz

$$p_{\text{th}}(N) = p_{\text{sus}} \left[1 - \left(1 - \frac{p_{\text{th}}(1)}{p_{\text{sus}}} \right) N^{-\gamma} \right], \quad (10)$$

to model the behaviour of the error threshold as a function of N , where γ and p_{sus} (the sustainable threshold) are parameters of the fit. For an error model with $\alpha = q/p = 1$, we find a sustainable threshold of $p_{\text{sus}} \approx 2.1\%$, with $\gamma = 1.06$ and $p_{\text{th}}(1) = 21.5\%$ (as shown in Fig. 1).

We observe that the decoder has a significantly higher tolerance to measurement noise as opposed to qubit noise, as shown in Fig. 1. We find that the maximum phase-flip error rate that the decoder can tolerate is $p \approx 2.9\%$ (for $q = 0$), compared with a maximum measurement error rate of $q \approx 8\%$ (for $p \rightarrow 0^+$). Our results show that the sweep decoder has an inbuilt resilience to measurement errors. To understand why this is the case, let us analyse the effect of measurement errors on the decoder. First, consider measurement errors that are far from phase-flip errors. A single isolated measurement error cannot cause the decoder to erroneously apply a Pauli- X operator. To deceive the decoder, two measurement errors must occur next to each other, such that a vertex becomes trailing. Second, measurement errors in the neighbourhood of phase-flip errors can interfere with the decoder, and prevent it from applying a correction. Thus, to affect the performance of the decoder, a single measurement error either has to occur close to a phase-flip measurement error. This explains why the sweep decoder has a higher tolerance of measurement errors relative to phase-flip errors.

We also simulated the performance of the decoder for lattices with boundaries. We consider toric codes defined the family of rhombic dodecahedral lattices with boundaries that we described earlier. We find that the sustainable threshold of toric codes defined on this lattice family is $p_{\text{sus}} \approx 2.1\%$, for an error model where $\alpha = q/p = 1$. This value matches the sustainable threshold for the corresponding lattice with periodic boundary conditions, as expected.

A natural question to ask is whether applying the sweep rule multiple times per stabilizer measurement improves the performance of the decoder. Multiple applications of the rule could be feasible if gates are much faster than measurements, as in e.g. in trapped-ion qubits⁵⁶. We found that increasing the number of applications of the rule per syndrome measurement significantly improved the performance of the decoder for a variety of error models (including error models where $\alpha = q/p > 1$); see Fig. 4 for an example.

The ability to change the sweep direction gives us parameters that we can use to tune the performance of the decoder. They are: the frequency with which we change the sweep direction and the order in which we change the sweep direction. We investigated the effect of varying both of these parameters. The most significant parameter is the direction-change frequency. Our simulation naturally divides into two phases: the error suppression phase where the rule applied while errors are happening, and the perfect decoding phase where the rule is applied without errors. In the error suppression phase, we want to prevent the build-up of errors near the boundaries so we anticipate that we may want to vary the sweep direction more frequently. We find that changing direction after $\sim \log L$ sweeps in the error-suppression phase and L sweeps in the perfect decoding phase gave the best performance, as shown in Fig. 5. We find that the order in which we change the sweep direction does not appreciably impact the performance of the decoder. In addition, we find that the performance of the regular sweep decoder is superior to the greedy sweep decoder introduced in³⁹.

Finally, we evaluated the performance of the sweep decoder against a simple correlated noise model, finding a reduced error threshold (see Supplementary Note 3).

Discussion

In this article, we extended the definition of the sweep rule CA to causal codes. We also proved that the sweep decoder has a non-zero error threshold for 3D toric codes defined on a family of lattices with boundaries. In addition, we benchmarked and optimized the decoder for various 3D toric codes.

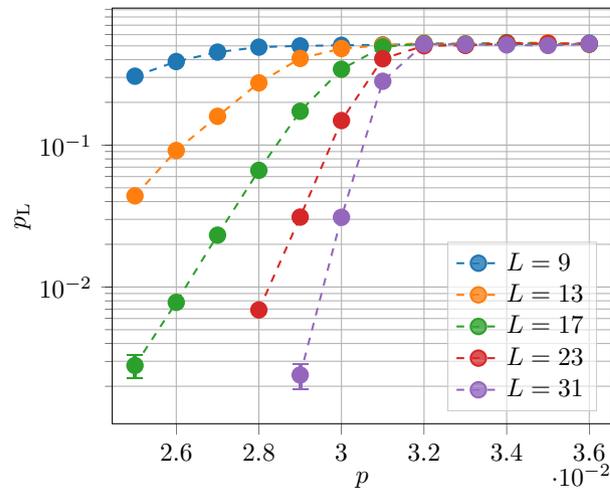


Figure 4. Improving the performance of the sweep decoder applied to the toric code on the rhombic dodecahedral lattice (with boundaries) by applying the rule multiple times per syndrome measurement. We set $\alpha = q/p = 1$ and we fix $N = 2^{10}$ error correction cycles. We plot the logical error rate p_L as a function of p for different linear lattice sizes L . We applied the rule three times per syndrome measurement and we observe an error threshold of $p_{th} \approx 3.2\%$, an improvement of over the corresponding error threshold of $p_{th} \approx 2.17\%$ when we applied the rule once per syndrome measurement (see the inset of Fig. 1a). We use 10^4 Monte Carlo samples for each point.

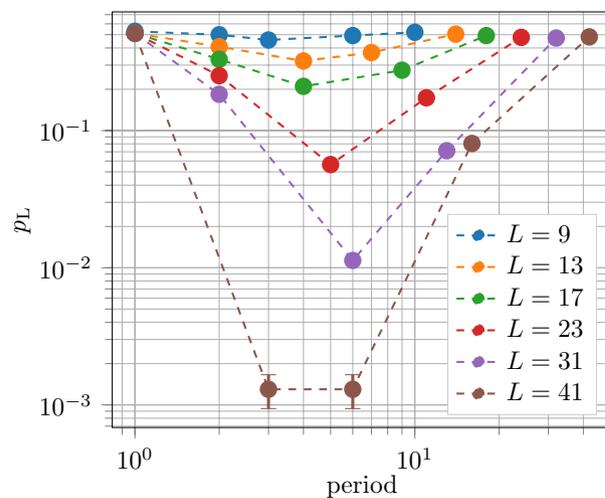


Figure 5. Optimizing the direction-change frequency for the sweep decoder applied to the toric code on the rhombic dodecahedral lattice (with boundaries). We plot the logical error rate p_L as a function of the direction-change period for various values of L . The number of error correction cycles is $N = 2^{10}$ and $p = q = 0.021$. We achieve the best performance when we change sweep direction every $\sim \log L$ cycles. We use 10^4 samples for each point.

We now comment on the performance of the sweep decoder compared with other decoding algorithms. Recall that in Table 1, we list the error thresholds obtained from numerical simulations for various decoders applied to toric codes defined on the cubic lattice subject to phase-flip and measurement noise (see Supplementary Note 2 for details on the sweep decoder numerics). Although the sweep decoder does not have the highest error threshold, it has other advantages that make it attractive. Firstly, as it is a CA decoder, it is highly parallelizable, which is an advantage when compared to decoding algorithms that require more involved processing such as the RG decoder introduced in Ref.⁵⁰. In addition, whilst neural network decoders such as⁴⁹ have low complexity once the network is trained, for codes with boundaries the training cost may scale exponentially with the code distance⁵⁷. Also, the sweep decoder is the only decoder in Table 1 that exhibits single-shot error correction. In contrast, using the RG decoder it is necessary to repeat the stabilizer measurements $O(L)$ times before finding a correction, which further complicates the decoding procedure. Finally, we note there is still a large gap between

the highest error thresholds in Table 1 and the theoretical maximum error threshold of $p_{\text{th}} = 23.180(4)\%$, as predicted by statistical mechanical mappings^{58–61}.

The sweep decoder could also be used in other topological codes with boundaries. Recently, it was shown that toric code decoders can be used to decode color codes⁶². Therefore, we could use the sweep decoder to correct errors in ($d \geq 3$)-dimensional color codes with boundaries, as long as the toric codes on the restricted lattices of the color codes are causal codes.

We expect that the sweep decoder would be well-suited to (just-in-time) JIT decoding, which arises when one considers realizing a 3D topological code using a constant-thickness slice of active qubits. In this context, the syndrome information from the full 3D code is not available, but we still need to find a correction. The sweep decoder is ideally suited to this task, because it only requires local information to find a correction. Let us consider the JIT decoding problem described in³² as an example. We start with a single layer of 2D toric code. Then we create a new 2D toric code layer above the first, with each qubit prepared in the $|0\rangle$ state. Next, we measure the X stabilizers of the 3D toric code that link the two 2D layers. Then, we measure the qubits of the lower layer in the Z basis. We iterate this procedure until we have realized the full 3D code.

The X stabilizer measurements will project two layers of 2D toric code into a single 3D toric code, up to a random Pauli Z error applied to the qubits. This error will have a syndrome consisting of closed loops of edges. The standard decoding strategy is simply to apply a Z operator whose boundary is equal to the closed loops. However, when measurement errors occur, the syndrome will consist of broken loops. We call the ends of broken loops breakpoints. To decode with measurement errors, one can first close the broken loops (pair breakpoints) using the minimum-weight perfect matching algorithm, before applying a Z correction. However, in the JIT scenario, some breakpoints may need to be paired with other breakpoints that will appear in the future. In³², Brown proposed deferring the pairing of breakpoints until later in the procedure, to reduce the probability of making mistakes. However, given the innate robustness of the sweep decoder, instead of using the ‘repair-then-correct’ decoder outlined above, we could simply apply the sweep decoder at every step and not worry about the measurement errors. Given our numerical results, we anticipate that the sweep decoder would be effective in this case, and so provides an alternative method of JIT decoding to that is worth exploring.

We finish this section by suggesting further applications of the sweep decoder, such as decoding more general quantum LDPC codes, e.g. homological product codes⁶³, especially those introduced in⁶⁴. The abstract reformulation of the sweep rule CA presented in the Introduction provides a clear starting point for this task. However, we emphasize that it is still uncertain as to whether the sweep decoder would work in this case, as more general LDPC codes may not share the properties of topological codes that are needed in the proof of Theorem 1. In addition, we would like to prove the existence of a non-zero error threshold when measurements are unreliable. From our investigation of this question, it seems that the proof technique in Supplementary Note 1 breaks down for this case.

Methods

In this section, we prove Lemma 1, which concerned the properties of the sweep rule in an abstract setting. In addition, we show that the sweep rule retains essentially the same properties for rhombic dodecahedral lattices with boundaries. We require these properties to prove a non-zero error threshold for the sweep decoder (see Supplementary Note 1). We begin with a useful lemma about causal diamonds, which we will use throughout this section.

Lemma 3 *Let V be a partially ordered set. For any finite subsets $U, W \subseteq V$, if $U \subseteq W$, then*

$$U \subseteq \diamond(U) \subseteq \diamond(W). \quad (11)$$

Proof We recall that $\diamond(U) = \uparrow(\inf U) \cap \downarrow(\sup U)$. As both $\uparrow(\inf U)$ and $\downarrow(\sup U)$ contain U , their intersection also contains U , i.e. $U \subseteq \diamond(U)$. If $U \subseteq W$, then $\inf W \leq \inf U$ so $\uparrow(\inf U) \subseteq \uparrow(\inf W)$. Likewise, if $U \subseteq W$, then $\sup U \leq \sup W$ and therefore $\downarrow(\sup U) \subseteq \downarrow(\sup W)$. Consequently, $\diamond(U) \subseteq \diamond(W)$. \square

Proof of sweep rule properties. Proof of Lemma 1 First, we prove the support property by induction. At time step $T = 1$ (before the rule is applied), this property holds. Now, consider the syndrome at time T , $\sigma^{(T)}$. Let $U^{(T)}$ denote the set of trailing locations of the syndrome at time step T . Between time steps T and $T + 1$, for each trailing location $u \in U^{(T)}$, the sweep rule will return a subset of qubits $\varphi^{(T)}(u)$ with the property that $[\partial\varphi^{(T)}(u)]|_u = \sigma^{(T)}|_u$. Therefore, the syndrome at time step $T + 1$ is

$$\sigma^{(T+1)} = \sigma^{(T)} + \sum_{u \in U^{(T)}} \partial\varphi^{(T)}(u). \quad (12)$$

By assumption, $\diamond(\partial\varphi(u)) = \diamond(\sigma^{(T)}|_u)$, and $\diamond(\sigma^{(T)}) \subseteq \diamond(\sigma)$. Making multiple uses of Lemma 3, we have

$$\sigma^{(T+1)} \subseteq \diamond(\sigma^{(T+1)}) = \diamond\left(\sigma^{(T)} \cup \bigcup_{u \in U^{(T)}} \partial\varphi^{(T)}(u)\right) \subseteq \diamond\left(\diamond(\sigma^{(T)}) \cup \bigcup_{u \in U^{(T)}} \diamond(\partial\varphi^{(T)}(u))\right) \subseteq \diamond(\sigma). \quad (13)$$

Next, we prove the propagation property, also by induction. The property is true at time step $T = 1$. Now, we prove the inductive step from time step $T - 1$ to T . As long as $\sigma^{(T)} \neq 0$, for every $S \in \sigma^{(T)}$, either $S \in \sigma^{(T-1)}$ or there exists an edge in the syndrome graph between the node corresponding to S and a node corresponding to $S' \in \sigma^{(T-1)}$. By invoking the triangle inequality, we conclude that

$$d_G(S, \sigma) \leq d_G(S, S') + d_G(S', \sigma) = 1 + (T - 1) = T. \tag{14}$$

To prove the removal property, we define a function

$$f_\sigma(T) = \max_{v \in \bigcup \sigma^{(T)}} \max_{N \in \mathcal{N}(v, \sup \sigma)} \ell(N). \tag{15}$$

In words, $f_\sigma(T)$ is the length of the longest chain between any location $v \in \bigcup \sigma^{(T)}$ and the supremum of the original syndrome, $\sup \sigma$. If $\sigma^{(T)} = \emptyset$, then we set $f_\sigma(T) = 0$. We now show that $f_\sigma(T)$ is a monotonically decreasing function of T . At time T , any location $v \in \bigcup \sigma^{(T)}$ that maximizes the value of $f(T)$ will necessarily be trailing. Between time steps T and $T + 1$, a subset of qubits $\varphi^{(T)}(v) \in \uparrow(v)$ will be returned, and the syndrome will be modified such that $v \notin \bigcup \sigma^{(T+1)}$. Instead, there will be new locations $u \in \bigcup \sigma^{(T+1)}$, where every $u \succ v$, which implies that $f_\sigma(T + 1) < f_\sigma(T)$. We note that because every qubit $Q \in \mathcal{Q}$ contains its unique infimum, it is impossible for a qubit to be returned multiple times by the rule in one time step.

The removal property immediately follows from the monotonicity of $f_\sigma(T)$. As we consider lattices with a finite number of locations, $f_\sigma(1) = \max_{v \in \bigcup \sigma} \max_{N \in \mathcal{N}(v, \sup \sigma)} \ell(N)$ will be finite. And between each time step, $f_\sigma(T)$ decreases by at least one, which implies that $\sigma^{(T)} = \emptyset$ for all $T > \max_{v \in \bigcup \sigma} \max_{N \in \mathcal{N}(v, \sup \sigma)} \ell(N)$. \square

Sweep rule properties for rhombic dodecahedral lattices with boundaries. In this section, we show that the sweep rule retains the support, propagation and removal properties for rhombic dodecahedral lattices with boundaries, with some minor modifications.

We recall that we use a set of eight sweep directions $\Omega = (\pm 1, \pm 1, \pm 1)$. By inspecting Fig. 3, one can verify that no $\vec{\omega} \in \Omega$ is perpendicular to any of the edges of the rhombic dodecahedral lattice, so the partial order is always well defined. In the Results, we neglected a subtlety concerning causal regions. Consider the causal region of $U \subseteq \mathcal{L}_0$ with respect to $\vec{\omega}$, $\mathcal{R}_{\vec{\omega}}(U) = 2^{\diamond_{\vec{\omega}}(U) \cap \mathcal{L}_0}$. For a given U , causal regions with respect to different sweep directions may not be the same. Therefore, we must modify the definition of the causal region. Let $\{\vec{\omega}_1, \vec{\omega}_2, \dots, \vec{\omega}_8\}$ be an ordering of the sweep directions $\vec{\omega}_j \in \Omega$. We recursively define the causal region of U to be

$$\mathcal{R}(U) = \mathcal{R}_{\vec{\omega}_8} \circ \dots \circ \mathcal{R}_{\vec{\omega}_2} \circ \mathcal{R}_{\vec{\omega}_1}(U), \tag{16}$$

i.e. to compute $\mathcal{R}(U)$, we take the causal region of U with respect to $\vec{\omega}_1$, then we take the causal region of $\mathcal{R}_{\vec{\omega}_1}(U)$ with respect to $\vec{\omega}_2$, and similarly until we reach $\vec{\omega}_8$.

The first step in showing that the sweep rule has the desired properties is to prove Lemma 2. This lemma is sufficient for proving the removal property of the rule.

Proof of Lemma 1 Any vertex not on the boundaries of \mathcal{L} satisfies the trailing location condition for all $\vec{\omega} \in \Omega$, so we only need to check the vertices on the boundaries.

First, we consider the rough boundaries. On each such boundary, there are vertices that do not satisfy the trailing vertex condition for certain sweep directions. Let us examine each rough boundary in turn. First, consider the vertices on the top rough boundary, an example of which is highlighted in Fig. 6a. For these vertices, the problematic sweep directions are those that point inwards, i.e. $\vec{\omega} = (x, y, -1)$, $x, y = \pm 1$. Similarly, the problematic sweep directions for the vertices on the bottom rough boundary are $\vec{\omega} = (x, y, 1)$, $x, y = \pm 1$. Next, consider the vertices on the right rough boundary (see Fig. 6b for an example). The problematic sweep directions for these vertices are also those that point inwards, i.e. $\vec{\omega} = (-1, y, z)$, $y, z = \pm 1$. Analogously, the problematic sweep directions for the vertices on the left rough boundary are $\vec{\omega} = (1, y, z)$, $y, z = \pm 1$. Therefore, the satisfying sweep directions for vertices on the rough boundaries are those that point outwards, as shown in Fig. 7.

Now, consider the smooth boundaries. We have already analysed the vertices which are part of a rough boundary and a smooth boundary. For certain sweep directions, some vertices in the bulk of the smooth boundary do not satisfy the trailing vertex condition because of a missing face (see Fig. 2d for an example). For each such vertex, there are two problematic directions (both of which point outwards from the relevant smooth boundary). In Fig. 2d, these directions are $\vec{\omega} = -(1, 1, 1)$ and $\vec{\omega}' = (1, -1, 1)$. However, for each smooth boundary there are four sweep directions (the ones that point inwards) for which every vertex in the bulk of the boundary satisfies the trailing vertex condition. Figure 7 illustrates the satisfying directions for each smooth boundary.

We recall that a local region of \mathcal{L} has diameter smaller than $L/2$. Any such region can intersect at most two rough boundaries and one smooth boundary e.g. the boundaries highlighted in Fig. 7a, Fig. 7c, and Fig. 7e. For this combination of boundaries, there is only one sweep direction for which all vertices satisfy the trailing vertex condition: $\vec{\omega} = (1, 1, 1)$. This sweep direction will also work for a local region that intersects any two of the above boundaries, and any region that intersects any one of the above boundaries. There are eight possible combinations of three boundaries that can be intersected by a local region (corresponding to the eight corners of the cubes in Fig. 7). Therefore, by symmetry, the set of sweep directions required for the Lemma to hold are exactly $\Omega = \{(\pm 1, \pm 1, \pm 1)\}$. \square

We now explain how to modify the proof of Lemma 1 such that it applies to our family of rhombic dodecahedral lattices with boundaries. First, we note that the proof of the support property is identical, except we replace $\diamond(\sigma)$ by $\mathcal{R}(\sigma)$. The proof of the propagation property is essentially the same, except that there may be some time steps where the syndrome does not move, as there are some sweep directions for which certain syndromes are immobile. However, this does not affect the upper bound on the propagation distance of any syndrome: it is still upper-bounded by T , the number of applications of the rule. We note that in Lemma 1, the propagation distance refers to path lengths in the syndrome graph. However, one can verify that there will always be an edge

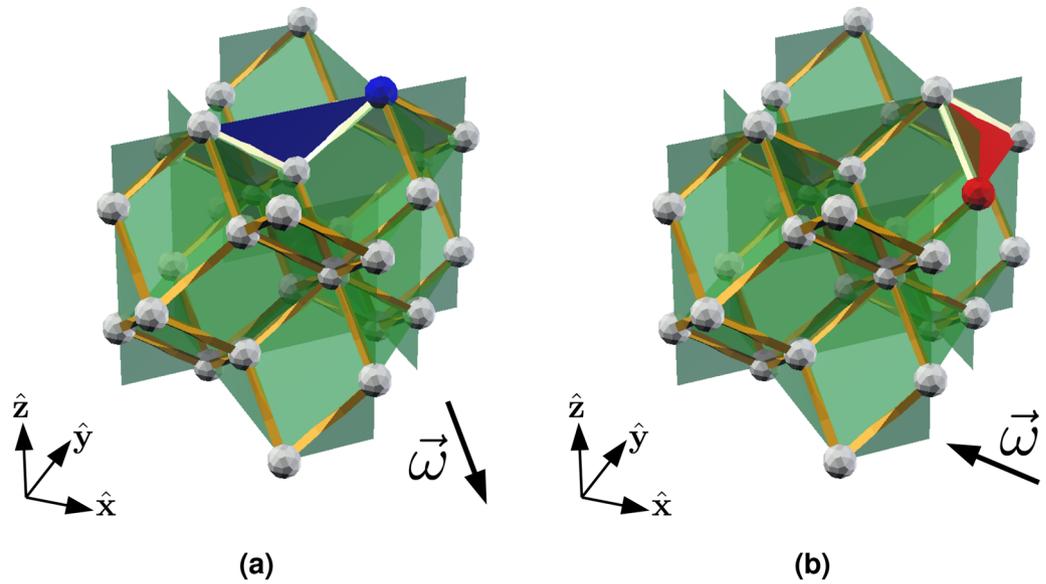


Figure 6. Vertices on the rough boundaries that do not satisfy the trailing vertex condition. In (a), we highlight such a vertex, v , in blue on the top rough boundary. Consider the highlighted qubit (blue face) and its syndrome (light yellow edges). For the sweep directions $\vec{\omega} = (1, -1, -1)$ and $\vec{\omega}' = (-1, 1, -1)$, v does not satisfy the trailing vertex condition because the blue face is not in $\uparrow(v)$. In (b), we highlight a vertex in red on the right rough boundary that does not satisfy the trailing vertex condition for the same reason as (a), where the problematic sweep direction is $\vec{\omega} = (-1, -1, 1)$.

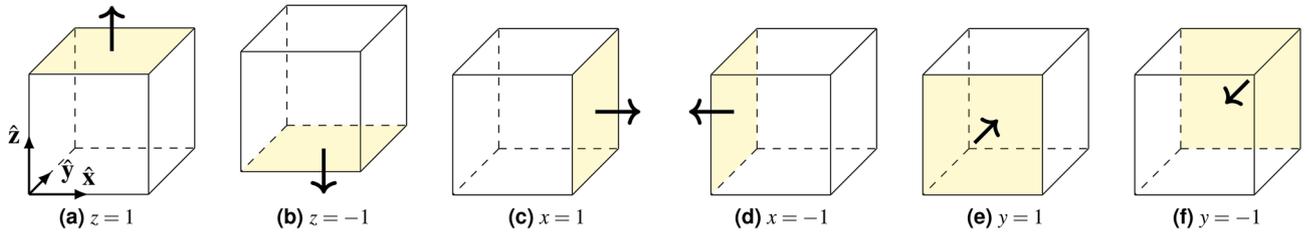


Figure 7. The constraints on the sweep directions $\vec{\omega} = (x, y, z) \in \{(\pm 1, \pm 1, \pm 1)\}$ such that all vertices on the highlighted boundary satisfy the trailing vertex condition.

linking any syndrome $\sigma^{(T)}$ and its corresponding syndrome at the following time step $\sigma^{(T+1)}$ in \mathcal{L} . Therefore, the upper bound on the propagation distance also applies to the distance defined as the length of the shortest path between two vertices of \mathcal{L} .

Finally, the removal property also holds for any syndrome σ with $\mathcal{R}(\sigma)$ contained in a local region, but with a longer required removal time. Suppose that we use the ordering $\{\vec{\omega}_1, \vec{\omega}_2, \dots, \vec{\omega}_8\}$ and apply the sweep rule for T^* time steps using $\vec{\omega}_1$, followed by T^* applications using $\vec{\omega}_2$, and so on until we reach $\vec{\omega}_8$, where

$$T^* = \max_{\vec{\omega} \in \Omega} \max_{(\inf \mathcal{R}(\sigma) \downarrow \sup \mathcal{R}(\sigma))} |(\inf \mathcal{R}(\sigma) \downarrow \sup \mathcal{R}(\sigma))|, \tag{17}$$

i.e. the longest causal path between the infimum and supremum of $\mathcal{R}(\sigma)$, maximized over the set of sweep directions.

By Lemma 2, there will always exist a sweep direction such that the trailing location condition is satisfied at every vertex in $\mathcal{R}(\sigma)$. Therefore, one can make a similar argument to the one made in the proof of Lemma 1 to show that there exists a monotone for at least one sweep direction, $\vec{\omega}$,

$$f(T) = \max_{v \in \sigma^{(T)}} |v \downarrow \sup \sigma^{(T)}|, \tag{18}$$

which decreases by one at each time step the sweep rule is applied with direction $\vec{\omega}$. By the support property $f_\sigma(T) \leq T^*$ for all T , so if we apply the rule T^* times with sweep direction $\vec{\omega}$, then the syndrome is guaranteed to be removed. But a priori we do not know which sweep direction(s) will remove a given syndrome. Therefore, to guarantee the removal of a syndrome, we must apply the sweep rule T^* times in each direction, which gives a total removal time of $|\Omega| \times T^*$.

Received: 1 June 2020; Accepted: 4 January 2021

Published online: 21 January 2021

References

- Shor, P. Fault-tolerant quantum computation. In *Proceedings of 37th Conference on Foundations of Computer Science* 56–65 (IEEE Comput. Soc. Press, 1996) <https://doi.org/10.1109/SFCS.1996.548464>.
- Preskill, J. Reliable quantum computers. *Proc. R. Soc. A* **454**, 385–410. <https://doi.org/10.1098/rspa.1998.0167> (1998).
- Campbell, E. T., Terhal, B. M. & Vuillot, C. Roads towards fault-tolerant universal quantum computation. *Nature* **549**, 172–179. <https://doi.org/10.1038/nature23460> (2017).
- Fowler, A. G., Mariantoni, M., Martinis, J. M. & Cleland, A. N. Surface codes: Towards practical large-scale quantum computation. *Phys. Rev. A* **86**, 032324. <https://doi.org/10.1103/PhysRevA.86.032324> (2012).
- Terhal, B. M. Quantum error correction for quantum memories. *Rev. Mod. Phys.* **87**, 307. <https://doi.org/10.1103/RevModPhys.87.307> (2015).
- Das, P. *et al.* A scalable decoder micro-architecture for fault-tolerant quantum computing. Preprint at [arXiv:2001.06598](https://arxiv.org/abs/2001.06598) (2020).
- Gottesman, D. Stabilizer Codes and Quantum Error Correction. Ph.D. thesis, Caltech (1997).
- Calderbank, A. R. & Shor, P. W. Good quantum error-correcting codes exist. *Phys. Rev. A* **54**, 1098–1105. <https://doi.org/10.1103/PhysRevA.54.1098> (1996).
- Kitaev, A. Y. Fault-tolerant quantum computation by anyons. *Ann. Phys.* **303**, 2–30. [https://doi.org/10.1016/S0003-4916\(02\)00018-0](https://doi.org/10.1016/S0003-4916(02)00018-0) (2003).
- Dennis, E., Kitaev, A., Landahl, A. & Preskill, J. Topological quantum memory. *J. Math. Phys.* **43**, 4452–4505. <https://doi.org/10.1063/1.1499754> (2002).
- Bombín, H. An introduction to topological quantum codes. In *Topological codes* (eds Lidar, D. A. & Brun, T. A.) (Cambridge University Press, Cambridge, 2013).
- Brown, B. J., Loss, D., Pachos, J. K., Self, C. N. & Wootton, J. R. Quantum memories at finite temperature. *Rev. Mod. Phys.* **88**, 045005. <https://doi.org/10.1103/RevModPhys.88.045005> (2016).
- Bombín, H. & Martin-Delgado, M. A. Topological computation without braiding. *Phys. Rev. Lett.* **98**, 160502. <https://doi.org/10.1103/PhysRevLett.98.160502> (2007).
- Bombín, H. Gauge color codes: Optimal transversal gates and gauge fixing in topological stabilizer codes. *New J. Phys.* **17**, 083002. <https://doi.org/10.1088/1367-2630/17/8/083002> (2015).
- Kubica, A. & Beverland, M. E. Universal transversal gates with color codes: A simplified approach. *Phys. Rev. A* **91**, 032330. <https://doi.org/10.1103/PhysRevA.91.032330> (2015).
- Kubica, A., Yoshida, B. & Pastawski, F. Unfolding the color code. *New J. Phys.* **17**, 083026. <https://doi.org/10.1088/1367-2630/17/8/083026> (2015).
- Webster, P. & Bartlett, S. D. Locality-preserving logical operators in topological stabilizer codes. *Phys. Rev. A* **97**, 012330. <https://doi.org/10.1103/PhysRevA.97.012330> (2018).
- Vasmer, M. & Browne, D. E. Three-dimensional surface codes: Transversal gates and fault-tolerant architectures. *Phys. Rev. A* **100**, 12312. <https://doi.org/10.1103/PhysRevA.100.12312> (2019).
- Bravyi, S. & König, R. Classification of topologically protected gates for local stabilizer codes. *Phys. Rev. Lett.* **110**, 170503. <https://doi.org/10.1103/PhysRevLett.110.170503> (2013).
- Pastawski, F. & Yoshida, B. Fault-tolerant logical gates in quantum error-correcting codes. *Phys. Rev. A* **91**, 13. <https://doi.org/10.1103/PhysRevA.91.012305> (2015).
- Jochym-O'Connor, T., Kubica, A. & Yoder, T. J. Disjointness of stabilizer codes and limitations on fault-tolerant logical gates. *Phys. Rev. X* **8**, 21047. <https://doi.org/10.1103/PhysRevX.8.021047> (2018).
- Barrett, S. D. & Kok, P. Efficient high-fidelity quantum computation using matter qubits and linear optics. *Phys. Rev. A* **71**, 2–5. <https://doi.org/10.1103/PhysRevA.71.060310> (2005).
- Nickerson, N. H., Fitzsimons, J. F. & Benjamin, S. C. Freely scalable quantum technologies using cells of 5-to-50 qubits with very lossy and noisy photonic links. *Phys. Rev. X* **4**, 1–17. <https://doi.org/10.1103/PhysRevX.4.041041> (2014).
- Monroe, C. *et al.* Large-scale modular quantum-computer architecture with atomic memory and photonic interconnects. *Phys. Rev. A* **89**, 1–16. <https://doi.org/10.1103/PhysRevA.89.022317> (2014).
- Kalb, N. *et al.* Entanglement distillation between solid-state quantum network nodes. *Science* **356**, 928–932. <https://doi.org/10.1126/science.aan0070> (2017).
- Kieling, K., Rudolph, T. & Eisert, J. Percolation, renormalization, and quantum computing with nondeterministic gates. *Phys. Rev. Lett.* **99**, 2–5. <https://doi.org/10.1103/PhysRevLett.99.130501> (2007).
- Raussendorf, R., Harrington, J. & Goyal, K. A fault-tolerant one-way quantum computer. *Ann. Phys.* **321**, 2242–2270. <https://doi.org/10.1016/j.aop.2006.01.012> (2006).
- Gimeno-Segovia, M., Shadbolt, P., Browne, D. E. & Rudolph, T. From three-photon Greenberger–Horne–Zeilinger states to ballistic universal quantum computation. *Phys. Rev. Lett.* **115**, 1–5. <https://doi.org/10.1103/PhysRevLett.115.020502> (2015).
- Rudolph, T. Why I am optimistic about the silicon-photonic route to quantum computing. *APL Photonics* **2**, 030901. <https://doi.org/10.1063/1.4976737> (2017).
- Nickerson, N. & Bombín, H. Measurement based fault tolerance beyond foliation. Preprint at [arXiv:1810.09621](https://arxiv.org/abs/1810.09621) (2018).
- Bombín, H. 2D quantum computation with 3D topological codes. Preprint at [arXiv:1810.09571](https://arxiv.org/abs/1810.09571) (2018).
- Brown, B. J. A fault-tolerant non-clifford gate for the surface code in two dimensions. *Sci. Adv.* <https://doi.org/10.1126/sciadv.aay4929> (2020).
- Harrington, J. Analysis of Quantum Error-Correcting Codes: Symplectic Lattice Codes and Toric Codes. Ph.D. thesis, Caltech. <https://doi.org/10.7907/AHMQ-EG82> (2004).
- Pastawski, F., Clemente, L. & Cirac, J. I. Quantum memories based on engineered dissipation. *Phys. Rev. A* **83**, 1–12. <https://doi.org/10.1103/PhysRevA.83.012304> (2011).
- Herold, M., Campbell, E. T., Eisert, J. & Kastoryano, M. J. Cellular-automaton decoders for topological quantum memories. *NPJ Quantum Inform.* <https://doi.org/10.1038/npjqi.2015.10> (2015).
- Herold, M., Kastoryano, M. J., Campbell, E. T. & Eisert, J. Cellular automaton decoders of topological quantum memories in the fault tolerant setting. *New J. Phys.* **19**, 1–12. <https://doi.org/10.1088/1367-2630/aa7099> (2017).
- Breuckmann, N. P., Duivenvoorden, K., Michels, D. & Terhal, B. M. Local decoders for the 2D and 4D toric code. *Quantum Inf. Comput.* **17**, 181–208. <https://doi.org/10.26421/QIC17.3-4> (2017).
- Dauphinais, G. & Poulin, D. Fault-tolerant quantum error correction for non-Abelian anyons. *Commun. Math. Phys.* **355**, 519–560. <https://doi.org/10.1007/s00220-017-2923-9> (2017).
- Kubica, A. & Preskill, J. Cellular-automaton decoders with provable thresholds for topological codes. *Phys. Rev. Lett.* **123**, 020501. <https://doi.org/10.1103/physrevlett.123.020501> (2019).
- Bombín, H. Single-shot fault-tolerant quantum error correction. *Phys. Rev. X* **5**, 031043. <https://doi.org/10.1103/PhysRevX.5.031043> (2015).

41. Campbell, E. T. A theory of single-shot error correction for adversarial noise. *Quantum Sci. Technol.* **4**, 025006. <https://doi.org/10.1088/2058-9565/aafc8f> (2019).
42. Wang, D. S., Fowler, A. G. & Hollenberg, L. C. Surface code quantum computing with error rates over 1%. *Phys. Rev. A* **83**, 2–5. <https://doi.org/10.1103/PhysRevA.83.020302> (2011).
43. Fowler, A. G. Minimum weight perfect matching of fault-tolerant topological quantum error correction in average $O(1)$ parallel time. *Quantum Inf. Comput.* **15**, 145–158 (2014).
44. Delfosse, N. & Nickerson, N. H. Almost-linear time decoding algorithm for topological codes. Preprint at [arXiv:1709.06218](https://arxiv.org/abs/1709.06218) (2017).
45. Bravyi, S. & Haah, J. Quantum self-correction in the 3D cubic code model. *Phys. Rev. Lett.* <https://doi.org/10.1103/PhysRevLett.111.200501> (2013).
46. Duclos-Cianci, G. & Poulin, D. Fast decoders for topological quantum codes. *Phys. Rev. Lett.* **104**, 1–5. <https://doi.org/10.1103/PhysRevLett.104.050504> (2010).
47. Duclos-Cianci, G. & Poulin, D. Fault-tolerant renormalization group decoder for abelian topological codes. *Quantum Inf. Comput.* **14**, 721–740 (2014).
48. Kubica, A. The ABCs of the Color Code: A Study of Topological Quantum Codes as Toy Models for Fault-Tolerant Quantum Computation and Quantum Phases Of Matter. Ph.D. thesis, Caltech. <https://doi.org/10.7907/059V-MG69> (2018).
49. Breuckmann, N. P. & Ni, X. Scalable neural network decoders for higher dimensional quantum codes. *Quantum* **2**, 68. <https://doi.org/10.22331/q-2018-05-24-68> (2018).
50. Duivenvoorden, K., Breuckmann, N. P. & Terhal, B. M. Renormalization group decoder for a four-dimensional toric code. *IEEE Trans. Inf. Theory* **65**, 2545–2562. <https://doi.org/10.1109/TIT.2018.2879937> (2019).
51. Alishious, A. B. & Sarvepalli, P. K. Decoding toric codes on three dimensional simplicial complexes. Preprint at [arXiv:1911.06056](https://arxiv.org/abs/1911.06056) (2019).
52. Alicki, R., Horodecki, M., Horodecki, P. & Horodecki, R. On thermal stability of topological qubit in Kitaev's 4d model. *Open Syst. Inf. Dyn.* **17**, 1–20. <https://doi.org/10.1142/S1230161210000023> (2010).
53. Gács, P. & Reif, J. A simple three-dimensional real-time reliable cellular array. *J. Comput. Syst. Sci.* **36**, 125–147. [https://doi.org/10.1016/0022-0000\(88\)90024-4](https://doi.org/10.1016/0022-0000(88)90024-4) (1988).
54. Van Den Berg, J. & Kesten, H. Inequalities with applications to percolation and reliability. *J. Appl. Probab.* **22**, 556–569. <https://doi.org/10.2307/3213860> (1985).
55. Vasmer, M. *Sweep-Decoder-Boundaries*. GitHub repository at <https://github.com/MikeVasmer/Sweep-Decoder-Boundaries> (2020).
56. Crain, S. *et al.* High-speed low-crosstalk detection of a 171Yb+ qubit using superconducting nanowire single photon detectors. *Commun. Phys.* **2**, 1–8. <https://doi.org/10.1038/s42005-019-0195-8> (2019).
57. Maskara, N., Kubica, A. & Jochym-O'Connor, T. Advantages of versatile neural-network decoding for topological codes. *Phys. Rev. A* **99**, 052351. <https://doi.org/10.1103/PhysRevA.99.052351> (2019).
58. Hasenbusch, M., Toldin, F. P., Pelissetto, A. & Vicari, E. Magnetic-glassy multicritical behavior of the three-dimensional J Ising model. *Phys. Rev. B* **76**, 184202. <https://doi.org/10.1103/PhysRevB.76.184202> (2007).
59. Ozeki, Y. & Ito, N. Multicritical dynamics for the J Ising model. *J. Phys. A* **31**, 5451–5461. <https://doi.org/10.1088/0305-4470/31/24/007> (1998).
60. Ohno, T., Arakawa, G., Ichinose, I. & Matsui, T. Phase structure of the random-plaquette Z₂ gauge model: Accuracy threshold for a toric quantum memory. *Nucl. Phys. B* **697**, 462–480. <https://doi.org/10.1016/j.nuclphysb.2004.07.003> (2004).
61. Kubica, A., Beverland, M. E., Brandão, F., Preskill, J. & Svore, K. M. Three-dimensional color code thresholds via statistical-mechanical mapping. *Phys. Rev. Lett.* **120**, 180501. <https://doi.org/10.1103/PhysRevLett.120.180501> (2018).
62. Kubica, A. & Delfosse, N. Efficient color code decoders in $d \geq 2$ dimensions from toric code decoders. Preprint at [arXiv:1905.07393](https://arxiv.org/abs/1905.07393) (2019).
63. Tillich, J. P. & Zemor, G. Quantum LDPC codes with positive rate and minimum distance proportional to the square root of the blocklength. *IEEE Trans. Inf. Theory* **60**, 1193–1202. <https://doi.org/10.1109/TIT.2013.2292061> (2014).
64. Zeng, W. & Pryadko, L. P. Higher-dimensional quantum hypergraph-product codes with finite rates. *Phys. Rev. Lett.* **122**, 230501. <https://doi.org/10.1103/PhysRevLett.122.230501> (2019).

Acknowledgements

M.V. thanks Niko Breuckmann, Padraic Calpin, Earl Campbell, and Tom Stace for valuable discussions. A.K. acknowledges funding provided by the Simons Foundation through the “It from Qubit” Collaboration. This research was supported in part by Perimeter Institute for Theoretical Physics. Research at Perimeter Institute is supported in part by the Government of Canada through the Department of Innovation, Science and Economic Development Canada and by the Province of Ontario through the Ministry of Colleges and Universities. Whilst at UCL, M.V. was supported by the EPSRC (grant number EP/L015242/1). D.E.B. acknowledges funding provided by the EPSRC (grant EP/R043647/1). We acknowledge use of UCL supercomputing facilities.

Author contributions

Concept conceived by all the authors. A.K. generalized the sweep rule. All authors contributed to the design of the sweep decoder. Analytic proofs by A.K. and M.V. Simulations designed and written by M.V., with input from D.E.B. and A.K. Data collected and analysed by M.V. Manuscript prepared by M.V. and A.K.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1038/s41598-021-81138-2>.

Correspondence and requests for materials should be addressed to M.V.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2021